

# On a Non-Context-Free Extension of PDL

Stefan Göller and Dirk Nowotka

*Institute for Formal Methods in Computer Science (FMI)  
University of Stuttgart, Germany*

---

## Abstract

Over the last 25 years, a lot of work has been done on seeking for decidable non-regular extensions of Propositional Dynamic Logic (PDL). Only recently, an expressive extension of PDL, allowing visibly pushdown automata (VPAs) as a formalism to describe programs, was introduced and proven to have a satisfiability problem complete for deterministic double exponential time. Lately, the VPA formalism was extended to so called  $k$ -phase multi-stack visibly pushdown automata ( $k$ -MVPAs). Similarly to VPAs, it has been shown that the language of  $k$ -MVPAs have desirable effective closure properties and that the emptiness problem is decidable. On the occasion of introducing  $k$ -MVPAs, it has been asked whether the extension of PDL with  $k$ -MVPAs still leads to a decidable logic. This question is answered negatively here. We prove that already for the extension of PDL with 2-phase MVPAs with two stacks satisfiability becomes  $\Sigma_1^1$ -complete.

*Key words:* Propositional Dynamic Logic, Visibly Pushdown Automata, Multi-Stack Visibly Pushdown Automata, Decidability, Satisfiability

---

## 1 Introduction

Propositional Dynamic Logic (PDL) is a modal logic introduced by Fischer and Ladner [1] which allows to reason about regular programs. In PDL, there are two syntactic entities: formulas, built from boolean and modal operators and interpreted as sets of worlds of a Kripke structure; and programs, built from the operators test, union, composition, and Kleene star and interpreted as binary relations in a Kripke structure. Thence, the occurring programs can be seen as a regular language over an alphabet that consists of tests and atomic programs. However, the mere usage of regular programs limits the

---

*Email address:* {goeller,nowotka}@fmi.uni-stuttgart.de (Stefan Göller and Dirk Nowotka).

expressiveness of PDL as for example witnessed by the set of executions of well-matched calls and returns of a recursive procedure, cf. [2]. Therefore, non-regular extensions of PDL have been studied quite extensively [2,3,4,5]. An extension of PDL by a class  $\mathcal{L}$  of languages means that in addition to regular languages also languages in  $\mathcal{L}$  may occur in modalities of formulas.

One interesting result on PDL extensions, among many others as summarized in [2], is that already the extension of PDL with the *single* language  $\{a^n b a^n \mid n \geq 1\}$  leads to an undecidable logic [3]. In contrast to this negative result, Harel and Raz proved that adding to PDL a *single* language accepted by a single-minded pushdown automaton yields a decidable logic [6]. A simple-minded pushdown automaton is a restricted pushdown automaton, where each input symbol determines the next control state, the stack operation and the stack symbol to be pushed, in case a push operation is performed. Generalizing this concept, Alur and Madhusudan proposed in [7] visibly pushdown languages which are defined as languages accepted by visibly pushdown automata (VPAs). A VPA is a pushdown automaton, where the stack operation is determined by the input in the following way; the alphabet is partitioned into letters that prompt a push, internal, or pop action, respectively. Note that it is well-known that visibly pushdown automata are strictly more powerful than simple-minded pushdown automata. Recently, also for the model of visibly pushdown languages, a PDL extension has been investigated by Löding, Lutz, and Serre [4]. They proved that satisfiability of this PDL extension is complete for deterministic double exponential time. Note that for this result, every visibly pushdown language occurring in a formula must be over the same partition of the alphabet.

Recently,  $k$ -phase multi-stack visibly pushdown automata ( $k$ -MVPAs), a natural extension of VPAs, have been introduced in [8]. A  $k$ -MVPA is an automaton equipped with  $n$  stacks where, again, the actions on the stacks are determined by the input, more precisely, every input symbol specifies on which stack a push or pop operation or whether an internal operation is done. Moreover, a  $k$ -MVPA is restricted to accept only words that can be obtained by concatenating at most  $k$  *phases*, where a phase is a sequence of input symbols that invoke pop actions from at most one stack. Note that  $k$ -MVPAs with one stack coincide with VPAs.

Due to the various effective closure properties and a decidable emptiness problem of the language class described by  $k$ -MVPAs, it is an interesting question to ask if the corresponding extension of PDL is still decidable. This question was raised in [8] and is answered negatively in this article. We prove  $\Sigma_1^1$ -completeness for this PDL extension. A  $\Sigma_1^1$  lower bound already holds, if we restrict ourselves to deterministic 2-MVPAs with two stacks. This is the weakest possible instance of  $k$ -MVPAs that is still more powerful than VPAs. Our proof relies on the same technique of the  $\Sigma_1^1$ -hardness proof of undecid-

ability of PDL extended with the single language  $\{a^n b a^n \mid n \geq 1\}$ , which is presented in [2]. Note however, that  $\{a^n b a^n \mid n \geq 1\}$  is not recognized by any  $k$ -MVPA for any  $k$ .

We proceed as follows. We recapitulate  $k$ -MVPAs in Section 2. Section 3 introduces the extension of PDL with  $k$ -MVPAs. A  $\Sigma_1^1$ -completeness proof is presented in Section 4.

## 2 $k$ -Phase Multi-Stack Visibly Pushdown Automata

In this section we recall the definition of  $k$ -phase multi-stack visibly pushdown automata from [8].

Let  $\mathbb{N} = \{0, 1, 2, \dots\}$  denote the natural numbers. Let  $n \in \mathbb{N}$ , then  $[n] = \{1, 2, \dots, n\}$ . Note that  $[0] = \emptyset$ . Let  $\varepsilon$  denote the *empty word*. For some  $n \in \mathbb{N}$  an  $n$ -stack call-return alphabet is a tuple  $\tilde{\Sigma}_n = \langle \{\Sigma_c^i, \Sigma_r^i\}_{i \in [n]}, \Sigma_{int} \rangle$  of pairwise disjoint finite alphabets. Let  $\Sigma_c = \bigcup_{i \in [n]} \Sigma_c^i$  and  $\Sigma_r = \bigcup_{i \in [n]} \Sigma_r^i$  for every  $i \in [n]$ , and let  $\Sigma = \Sigma_c \cup \Sigma_r \cup \Sigma_{int}$ . Let us fix  $\tilde{\Sigma}_n$  for the rest of this section.

**Definition 1** A multi-stack visibly pushdown automaton (MVPA) over  $\tilde{\Sigma}_n$  is a tuple  $M = (Q, Q_I, \Gamma, \delta, Q_F)$ , where (i)  $Q$  is a finite set of states, (ii)  $Q_I \subseteq Q$  is the set of initial states, (iii)  $\Gamma$  is a finite stack alphabet with  $\perp \in \Gamma \setminus \Sigma$ , (iv)  $\delta \subseteq (Q \times \Sigma_c \times Q \times \Gamma \setminus \{\perp\}) \cup (Q \times \Sigma_r \times \Gamma \times Q) \cup (Q \times \Sigma_{int} \times Q)$ , and (v)  $Q_F \subseteq Q$  is the set of final states.

A  $k$ -MVPA is *deterministic*, if  $|Q_I| = 1$  and for each  $q \in Q$ , for each  $a \in \Sigma$ , and for each  $\gamma \in \Gamma$  we have

$$\left| \delta \cap \left( \{q\} \times \{a\} \times (Q \times \Gamma \setminus \{\perp\} \cup \{\gamma\} \times Q \cup Q) \right) \right| \leq 1.$$

The set of *stacks* is defined as  $St = (\Gamma \setminus \{\perp\})^* \cdot \{\perp\}$ . A *configuration* of an MVPA is a pair  $(q, C)$  where  $q \in Q$  and  $C : [n] \rightarrow St$  is a mapping. A *run* of  $M$  on an input  $w = a_1 a_2 \dots a_m \in \Sigma^*$  ( $m \geq 0$ ), with  $a_i \in \Sigma$  for each  $i \in [m]$ , is a sequence of configurations  $(q_0, C_0)(q_1, C_1) \dots (q_m, C_m)$  such that

- $q_0 \in Q_I$  and  $C_0(i) = \perp$  for each  $i \in [n]$  and
- for every  $j \geq 1$  we have,
  - whenever  $a_j \in \Sigma_c^i$  for some  $i \in [n]$ , then there exists some  $\gamma \in \Gamma \setminus \{\perp\}$  such that  $(q_{j-1}, a_j, q_j, \gamma) \in \delta$ , and  $C_j(i) = \gamma \cdot C_{j-1}(i)$  and  $C_j(i') = C_{j-1}(i')$  for all  $i' \in [n]$  with  $i' \neq i$ ,
  - whenever  $a_j \in \Sigma_r^i$  for some  $i \in [n]$ , then there exists some  $\gamma \in \Gamma$  such that  $(q_{j-1}, a_j, \gamma, q_j) \in \delta$ , and  $C_j(i') = C_{j-1}(i')$  for all  $i' \in [n]$  with  $i' \neq i$  and

- either (i)  $\gamma = \perp$  and  $C_j(i) = C_{j-1}(i) = \perp$  or (ii)  $\gamma \neq \perp$  and  $C_{j-1}(i) = \gamma \cdot C_j(i)$ , and
- whenever  $a_j \in \Sigma_{int}$ , then  $(q_{j-1}, a_j, q_j) \in \delta$  and  $C_j(i) = C_{j-1}(i)$  for all  $i \in [n]$ .

We call a run  $(q_0, C_0)(q_1, C_1) \cdots (q_m, C_m)$  *accepting*, if  $q_m \in Q_F$ . Furthermore, we denote by  $L(M) = \{w \in \Sigma^* \mid \text{there exists an accepting run of } M \text{ on } w\}$  the *language* of  $M$ . A word  $w \in \Sigma^*$  is a *phase*, if  $w \in (\Sigma_c \cup \Sigma_{int} \cup \Sigma_r^i)^*$  for some  $i \in \mathbb{N}$ . For  $k \geq 1$ , we say a word is a *k-phase* if it can be obtained by concatenating at most  $k$  phases.

**Definition 2** A *k-phase multi-stack visibly pushdown automaton* (*k-MVPA*)  $M$  is a multi-stack visibly pushdown automaton that is restricted to accept *k-phases only*. Formally, we define

$$L(M) = \{w \in \Sigma^* \mid w \text{ is a } k\text{-phase and there exists an accepting run of } M \text{ on } w\}.$$

Note that  $n = 0$  implies that a *k-MVPA* is as powerful as a finite state automaton. Moreover, we get precisely the VPAs as introduced in [7] when  $n = 1$ .

### 3 Propositional Dynamic Logic over *k-MVPAs*

Fix some countable set  $\mathbb{P}$  of *atomic propositions*, and some  $k, n \in \mathbb{N}$  with  $k \geq 1$ . The *set of formulas*  $\Phi$  and the set of *tests* Tests of the logic  $\text{PDL}(k, n)$  over some  $n$ -stack call-return alphabet  $\tilde{\Sigma}_n = \langle \{\Sigma_c^i, \Sigma_r^i\}_{i \in [n]}, \Sigma_{int} \rangle$  are the smallest sets that satisfy the following conditions:

- $\mathbf{true} \in \Phi$ ,
- if  $p \in \mathbb{P}$ , then  $p \in \Phi$ ,
- if  $\varphi_1, \varphi_2 \in \Phi$ , then  $\varphi_1 \vee \varphi_2, \neg \varphi_1 \in \Phi$ ,
- if  $\varphi \in \Phi$ , then  $\varphi? \in \text{Tests}$
- if  $\varphi \in \Phi$  and  $\Psi \subset \text{Tests}$  is finite, then  $\langle \chi \rangle \varphi \in \Phi$ , where  $\chi$  is either a regular expression over  $\Sigma \cup \Psi$  or  $\chi$  is a *k-MVPA* over  $\langle \{\Sigma_c^i, \Sigma_r^i\}_{i \in [n]}, \Sigma_{int} \cup \Psi \rangle$ .

We introduce the usual abbreviations  $\mathbf{false} = \neg \mathbf{true}$ ,  $\varphi_1 \wedge \varphi_2 = \neg(\neg \varphi_1 \vee \neg \varphi_2)$ , and  $[\chi]\varphi = \neg \langle \chi \rangle \neg \varphi$ . A *Kripke structure* is a tuple  $K = (X, \{\rightarrow_a\}_{a \in \Sigma}, \rho)$ , where  $X$  is a set of *worlds*,  $\rightarrow_a \subseteq X \times X$  is a binary relation for each  $a \in \Sigma$ , and  $\rho : X \rightarrow 2^{\mathbb{P}}$  assigns to each world a set of atomic propositions. For each  $\varphi \in \Phi$  and for each  $w \in (\Sigma \cup \text{Tests})^*$ , define the binary relation  $\llbracket w \rrbracket_K \subseteq X \times X$  and the set  $\llbracket \varphi \rrbracket_K \subseteq X$  via mutual induction as follows:

- $\llbracket \varepsilon \rrbracket_K = \{(x, x) \mid x \in X\}$ ,

- if  $\varphi? \in \text{Tests}$ , then  $\llbracket \varphi? \rrbracket_K = \{(x, x) \mid x \in X \wedge x \in \llbracket \varphi \rrbracket_K\}$ ,
- if  $a \in \Sigma$ , then  $\llbracket a \rrbracket_K = \rightarrow_a$ ,
- if  $w \in (\Sigma \cup \text{Tests})^*$  and  $\tau \in \Sigma \cup \text{Tests}$ , then  $\llbracket w\tau \rrbracket_K = \llbracket w \rrbracket_K \circ \llbracket \tau \rrbracket_K$ ,
- if  $p \in \mathbb{P}$ , then  $\llbracket p \rrbracket_K = \{x \in X \mid p \in \rho(x)\}$ ,
- $\llbracket \varphi_1 \vee \varphi_2 \rrbracket_K = \llbracket \varphi_1 \rrbracket_K \cup \llbracket \varphi_2 \rrbracket_K$ ,
- $\llbracket \neg \varphi \rrbracket_K = X \setminus \llbracket \varphi \rrbracket_K$ ,
- $\llbracket \langle \chi \rangle \varphi \rrbracket_K = \{x \in X \mid \exists y \in X \exists w \in L(\chi) : (x, y) \in \llbracket w \rrbracket_K \wedge y \in \llbracket \varphi \rrbracket_K\}$ .

Note that since we restrict  $k$ -MVPAs to accept  $k$ -phases only, we additionally allow formulas of the kind  $\langle \alpha \rangle \varphi$ , where  $\alpha$  is a regular expression over a finite subset of  $\Sigma \cup \text{Tests}$ . A  $k$ -MVPA can accept a regular language over  $k$ -phases only, that is, not even  $\Sigma^*$  (if  $\Sigma$  contains two pop symbols from different stacks) can be recognized. However, since we would like to increase the expressiveness of PDL beyond regular programs, we have to explicitly take in regular expressions. If  $L$  is a language over a finite subset of  $\Sigma \cup \text{Tests}$ , we define  $\llbracket L \rrbracket_K = \bigcup_{w \in L} \llbracket w \rrbracket_K$ . In the following, we will write  $\langle L \rangle \varphi$  ( $\llbracket L \rrbracket \varphi$ ) instead of  $\langle \chi \rangle \varphi$  ( $\llbracket \chi \rrbracket \varphi$ ), where  $L$  is the language of  $\chi$  and  $\chi$  is either some regular expression or some  $k$ -MVPA. We also write  $(K, x) \models \varphi$  whenever  $x \in \llbracket \varphi \rrbracket_K$ . We say that  $K$  is a *model* for  $\varphi$ , if  $(K, x) \models \varphi$  for some world  $x$  of  $K$ . We say a  $\text{PDL}(k, n)$  formula  $\varphi$  is *satisfiable*, if there exists a model for  $\varphi$ . The *satisfiability problem* asks, given a  $\text{PDL}(k, n)$  formula  $\varphi$ , whether  $\varphi$  is satisfiable.

When restricting all automata that occur in a formula to be visibly push-down automata (i.e. over a single stack), Löding, Lutz and Serre obtained the following result:

**Theorem 3** ([4]) *Satisfiability of  $\text{PDL}(1, 1)$  is complete for deterministic double exponential time.*

#### 4 $\Sigma_1^1$ -Completeness of $\text{PDL}(k, n)$

For the  $\Sigma_1^1$  upper bound, we can easily adapt the proof of Proposition 9.4 in [2] and show that every satisfiable  $\text{PDL}(k, n)$  formula has a countable tree model. Thus, we can write down an existential second-order number-theoretic formula over  $\mathbb{N}$  that is valid if and only if  $\varphi$  is satisfiable.

For the lower bound, we prove that  $\text{PDL}(k, n)$  is  $\Sigma_1^1$ -hard already for  $k = 2$  and  $n = 2$ , i.e. we can restrict all occurring MVPAs to have 2 stacks and to accept 2-phases only. For this, we reduce the  $\Sigma_1^1$ -hard recurring tiling problem of the first quadrant of the plane to satisfiability of  $\text{PDL}(2, 2)$ . A *recurring tiling system*  $\mathcal{T} = (T, H, V, t_0)$  consists of a finite set of *tile types*  $T$ , a *horizontal matching relation*  $H \subseteq T \times T$ , a *vertical matching relation*  $V \subseteq T \times T$ , and a tile type  $t_0 \in T$ . A *solution for  $\mathcal{T}$*  is a mapping  $\mu : \mathbb{N} \times \mathbb{N} \rightarrow T$  such that

for infinitely many  $m \in \mathbb{N}$  we have  $\mu(0, m) = t_0$  and for all  $(n, m) \in \mathbb{N} \times \mathbb{N}$  we have

- if  $\mu(n, m) = t$  and  $\mu(n + 1, m) = t'$ , then  $(t, t') \in H$ , and
- if  $\mu(n, m) = t$  and  $\mu(n, m + 1) = t'$ , then  $(t, t') \in V$ .

The *recurring tiling problem* is to decide whether a given recurring tiling system has a solution.

**Theorem 4** ([9]) *The recurring tiling problem is  $\Sigma_1^1$ -complete.*

For the rest of the section fix some tiling system  $\mathcal{T} = (T, H, V, t_0)$ . Our goal is to translate  $\mathcal{T}$  into a PDL(2, 2) formula  $\varphi = \varphi(\mathcal{T})$  over the set of atomic propositions  $T$  such that  $\mathcal{T}$  has a solution if and only if  $\varphi$  is satisfiable.

Fix the 2-stack alphabet  $\tilde{\Sigma}_2 = \langle \{\Sigma_c^i, \Sigma_r^i\}_{i \in \{1, 2\}}, \Sigma_{int} \rangle$  where  $\Sigma_c^i = \{a_i\}$  and  $\Sigma_r^i = \{b_i\}$  for each  $i \in [2]$  and where  $\Sigma_{int} = \{c, d\}$ . Define the languages  $L_\ell$ ,  $L_\ell^\leftrightarrow$ , and  $L_\ell^\uparrow$  for each  $\ell \in \{0, 1\}$  as follows, where  $w_0 = a_1 b_2$  and  $w_1 = a_2 b_1$  and  $e_0 = d$  and  $e_1 = c$ :

$$\begin{aligned} L_\ell &= \{w_\ell^i e_\ell w_{1-\ell}^j e_{1-\ell} \mid i, j \geq 0 \text{ and } j \neq i + 1\}, \\ L_\ell^\leftrightarrow &= \{w_\ell^i e_\ell w_{1-\ell}^{i+\ell+1} \mid i \geq 0\}, \\ L_\ell^\uparrow &= \{w_\ell^i e_\ell w_{1-\ell}^{i-\ell+2} \mid i \geq 0\}. \end{aligned}$$

**Proposition 5** *For each of the languages  $L_\ell$ ,  $L_\ell^\leftrightarrow$ , and  $L_\ell^\uparrow$ , with  $\ell \in \{0, 1\}$ , there exists a deterministic 2-MVPA over  $\tilde{\Sigma}_2$  that accepts it.*

**PROOF.** Figures 1 to 3 depict 2-MVPAs recognizing  $L_\ell$ ,  $L_\ell^\leftrightarrow$ , and  $L_\ell^\uparrow$ , respectively, for  $\ell = 0$ . The case  $\ell = 1$  is deduced by simultaneously substituting  $a_1$ ,  $b_2$ ,  $c$ , and  $d$  by  $a_2$ ,  $b_1$ ,  $d$ , and  $c$ , respectively. Note that all automata are deterministic.  $\square$

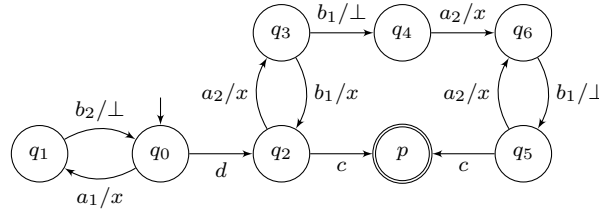


Fig. 1. A 2-MVPA recognizing  $L_0 = \{(a_1 b_2)^i d (a_2 b_1)^j c \mid i, j \geq 0, j \neq i + 1\}$ .

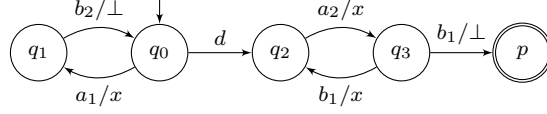


Fig. 2. A 2-MVPA recognizing  $L_0^{\leftrightarrow} = \{(a_1 b_2)^i d (a_2 b_1)^{i+1} \mid i \geq 0\}$ .

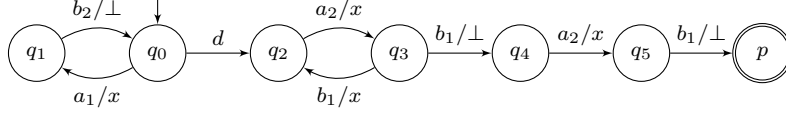


Fig. 3. An automaton recognizing  $L_0^{\downarrow} = \{(a_1 b_2)^i d (a_2 b_1)^{i+2} \mid i \geq 0\}$ .

Let  $\varphi_{\text{snake}}$  be defined as follows:

$$\begin{aligned} \varphi_{\text{snake}} = & \langle ca_1 b_2 d (a_2 b_1)^2 c \rangle \text{true} \wedge [\Sigma^* c] (\langle (a_1 b_2)^* d \rangle \text{true} \wedge [L_0] \text{false}) \\ & \wedge [\Sigma^* d] (\langle (a_2 b_1)^* c \rangle \text{true} \wedge [L_1] \text{false}). \end{aligned}$$

A *snake* of a Kripke structure  $K$  is an infinite path in  $K$  that is labeled by

$$c(a_1 b_2)^1 d(a_2 b_1)^2 c(a_1 b_2)^3 d(a_2 b_1)^4 c(a_1 b_2)^5 d(a_2 b_1)^6 c \dots$$

**Proposition 6** *Every model of  $\varphi_{\text{snake}}$  has a snake.*

**PROOF.** Let  $K = (X, \{\rightarrow_a\}_{a \in \Sigma}, \rho)$  be a model of  $\varphi_{\text{snake}}$ , i.e.  $(K, x) \models \varphi_{\text{snake}}$  for some  $x \in X$ . By the first conjunct of  $\varphi_{\text{snake}}$ , there exist worlds  $x_1, x_2 \in X$  such that  $(x, x_1) \in \llbracket ca_1 b_2 d \rrbracket_K$ , and  $(x_1, x_2) \in \llbracket (a_2 b_1)^2 c \rrbracket_K$ . Firstly, observe that  $(K, x_2) \models \langle (a_1 b_2)^+ d \rangle \text{true}$  by the third conjunct of  $\varphi_{\text{snake}}$ . This implies that  $(x_2, x_3) \in \llbracket (a_1 b_2)^i d \rrbracket_K$  for some  $x_3 \in X$  and some  $i \in \mathbb{N}$ . But clearly  $i = 3$ , for otherwise  $(K, x_1) \not\models [L_1] \text{false}$ . Thus we get  $(x_2, x_3) \in \llbracket (a_1 b_2)^3 d \rrbracket_K$ . Symmetrically, since  $(K, x_3) \models \langle (a_2 b_1)^+ c \rangle \text{true}$  and  $(K, x_2) \models [L_0] \text{false}$  by the second conjunct of  $\varphi_{\text{snake}}$ , there exists a world  $x_4$  such that  $(x_3, x_4) \in \llbracket (a_2 b_1)^4 c \rrbracket_K$ . By repeatedly applying the above argument, it is straightforward to see that there exists an infinite sequence of worlds  $x_1, x_2, x_3, x_4, \dots$  such that for each  $i \geq 1$  we have  $(x_{2i-1}, x_{2i}) \in \llbracket (a_2 b_1)^{2i} c \rrbracket_K$  and also  $(x_{2i}, x_{2i+1}) \in \llbracket (a_1 b_2)^{2i+1} d \rrbracket_K$ . Since additionally we have  $(x, x_1) \in \llbracket c(a_1 b_2)^1 d \rrbracket_K$ , there exists a snake in  $K$ .  $\square$

Let the programs  $\pi_{\downarrow\uparrow}$  and  $\beta$  and the formula  $\varphi_{\text{recur}}$  be defined as follows:

$$\begin{aligned} \pi_{\downarrow\uparrow} &= (a_1 b_2)^* d (a_2 b_1)^* c, \\ \beta &= \pi_{\downarrow\uparrow}^* \left( a_1 b_2 (t_0?) \pi_{\downarrow\uparrow} \cup (a_1 b_2)^* d (a_2 b_1)^* (t_0?) c \right), \\ \varphi_{\text{recur}} &= [\Sigma^* c] \langle \beta \rangle \text{true}. \end{aligned}$$

We call a world  $y$  on a snake  $\sigma$  *first column*, if either  $x_1 \xrightarrow{c} x_2 \xrightarrow{a_1} x_3 \xrightarrow{b_2} y$  or  $y \xrightarrow{c} x$  is a subpath of  $\sigma$ .

**Proposition 7** *Every model of  $\varphi_{\text{snake}} \wedge \varphi_{\text{recur}}$  has a snake on which infinitely often first column worlds satisfy the atomic proposition  $t_0$ .*

**PROOF.** Let  $K$  be a model of  $\varphi_{\text{snake}} \wedge \varphi_{\text{recur}}$ . By Proposition 6, there exists a snake  $\sigma_0$  in  $K$ . Fix an arbitrary world  $x_0$  on  $\sigma_0$  such that for some  $x \in X$  we have that  $x \xrightarrow{c}_K x_0$  is a subpath of  $\sigma_0$ . It is not hard to see that, by definition of  $\varphi_{\text{recur}}$  and by similar arguments as in the proof of Proposition 6, there exists a snake  $\sigma_1$  whose initial part agrees with  $\sigma_0$  up to world  $x_0$  and such that for some world  $x'_0$  on  $\sigma_1$ , we have  $(x_0, x'_0) \in \llbracket \beta \rrbracket_K$ . Moreover, by definition of  $\beta$ , on the subpath of  $\sigma_1$  from  $x_0$  to  $x'_0$  there exists some first column world that satisfies the atomic proposition  $t_0$ . Fix an arbitrary world  $x_1$  on  $\sigma_1$  such that there is a subpath from  $x'_0$  to  $x_1$  on  $\sigma_1$  such that additionally for some  $x' \in X$  we have that  $x' \xrightarrow{c}_K x_1$  is a subpath of  $\sigma_1$ . Again, we have  $(K, x_1) \models \langle \beta \rangle \text{true}$ . Hence again, there exists some snake  $\sigma_2$  whose initial part agrees with  $\sigma_1$  up to  $x_1$  such that for some world  $x'_1$  on  $\sigma_2$  we have  $(x_1, x'_1) \in \llbracket \beta \rrbracket_K$  and on the subpath of  $\sigma_2$  from  $x_1$  to  $x'_1$  some first column world of  $\sigma_2$  satisfies  $t_0$ . By repeatedly applying the same argument, we obtain a snake in  $K$  on which infinitely often first column worlds satisfy  $t_0$ .  $\square$

Let us now give a formula  $\varphi_{\text{tile}}$  that guarantees that every (reachable) world contains exactly one tile type:

$$\varphi_{\text{tile}} = [\Sigma^*] \left( \bigvee_{t \in T} \left( t \wedge \bigwedge_{t' \in T: t \neq t'} \neg(t \wedge t') \right) \right)$$

Next, we give a formula  $\varphi_{\updownarrow}^{\uparrow}$  that ensures that the types of vertically (horizontally) connected tiles satisfy the vertical (horizontal) matching relation:

$$\begin{aligned} \varphi_{\updownarrow}^{\uparrow} = & [\Sigma^* c(a_1 b_2)^+] \bigwedge_{t \in T} t \rightarrow \left( [L_0^{\leftarrow}] \bigvee_{t' \in T: (t, t') \in H} t' \wedge [L_0^{\uparrow}] \bigvee_{t' \in T: (t, t') \in V} t' \right) \wedge \\ & [\Sigma^* d(a_2 b_1)^+] \bigwedge_{t \in T} t \rightarrow \left( [L_1^{\leftarrow}] \bigvee_{t' \in T: (t, t') \in H} t' \wedge [L_1^{\uparrow}] \bigvee_{t' \in T: (t, t') \in V} t' \right) \end{aligned}$$

Our final formula  $\varphi$  is

$$\varphi = \varphi_{\text{snake}} \wedge \varphi_{\text{recur}} \wedge \varphi_{\text{tile}} \wedge \varphi_{\updownarrow}^{\uparrow}.$$

Before proving that  $\mathcal{T}$  has a solution if and only if  $\varphi$  is satisfiable, we introduce some more notation. Let  $A = \{(i, j) \in \mathbb{N} \times \mathbb{N} \mid 0 \leq j \leq i\}$ . We define a bijection



$\pi : \mathbb{N} \times \mathbb{N} \rightarrow A$  for all  $(n, m) \in \mathbb{N} \times \mathbb{N}$  as follows

$$\pi(n, m) = (n + m, m).$$

Thus,  $\pi^{-1}(i, j) = (i - j, j)$  for all  $(i, j) \in A$ .

**Lemma 8** *The recurring tiling system  $\mathcal{T}$  has a solution if and only if  $\varphi$  is satisfiable.*

**PROOF.**

*only-if:* Assume that  $\mathcal{T}$  has a solution  $\mu : \mathbb{N} \times \mathbb{N} \rightarrow T$ . Figure 4 depicts a model  $K = K(\mathcal{T}) = (X, \{\rightarrow_a\}_{a \in \Sigma}, \rho)$  that we can construct from  $\mathcal{T}$ . To all those worlds that are pictured by bullets, the mapping  $\rho$  assigns an arbitrary singleton subset from  $T$ . For the worlds  $x_{i,j}$ , where  $(i, j) \in A$ , we define

$$\rho(x_{i,j}) = \mu(\pi^{-1}(i, j)).$$

Thus, the world  $x_{i,j}$  represents the unique the pair  $(n, m) \in \mathbb{N}$  such that  $\pi(n, m) = (i, j)$ . It is straightforward to verify that  $(K, x) \models \varphi$ .

*if:* Let  $K = (X, \{\rightarrow_a\}_{a \in \Sigma}, \rho)$  be a model of  $\varphi$ , i.e. we have  $(K, x) \models \varphi$  for some world  $x \in X$ . We prove that  $\mathcal{T}$  has a solution. By Proposition 7 there exists a snake  $\sigma$  in  $K$  on which infinitely often first column worlds satisfy the atomic proposition  $t_0$ , since both  $\varphi_{\text{snake}}$  as well as  $\varphi_{\text{recur}}$  occur in  $\varphi$  as a conjunct and  $K$  is a model of  $\varphi$ . Recall that  $A = \{(i, j) \in \mathbb{N} \times \mathbb{N} \mid 0 \leq j \leq i\}$ . For each  $(i, j) \in A$ , fix some world  $x_{i,j}$  on  $\sigma$  such that  $x \xrightarrow{ca_1b_2}_K x_{0,0}$  and the following holds for each  $r \in \mathbb{N}$ :

$$x_{2r,0} \xrightarrow{da_2b_1}_K x_{2r+1,0} \quad \text{and} \quad x_{2r,s} \xrightarrow{a_1b_2}_K x_{2r,s-1}$$

and

$$x_{2r+1,2r+1} \xrightarrow{ca_1b_2}_K x_{2r+2,2r+2} \quad \text{and} \quad x_{2r+1,s} \xrightarrow{a_2b_1}_K x_{2r+1,s+1}$$

for all  $0 \leq s \leq 2r$ . Note that the first column nodes of  $\sigma$  are precisely the nodes  $\{x_{m,m} \mid m \in \mathbb{N}\}$ . Moreover, for all  $(2r, s), (2r+1, s) \in A$  we have

$$(x_{2r,s}, x_{2r+1,s}) \in \llbracket (a_1b_2)^{2r-s}d(a_2b_1)^{2r-s+1} \rrbracket_K, \quad (1)$$

$$(x_{2r+1,s}, x_{2r+2,s}) \in \llbracket (a_2b_1)^{2r-s+1}c(a_1b_2)^{2r-s+3} \rrbracket_K, \quad (2)$$

$$(x_{2r,s}, x_{2r+1,s+1}) \in \llbracket (a_1b_2)^{2r-s}d(a_2b_1)^{2r-s+2} \rrbracket_K, \quad (3)$$

$$(x_{2r+1,s}, x_{2r+2,s+1}) \in \llbracket (a_2b_1)^{2r-s+1}c(a_1b_2)^{2r-s+2} \rrbracket_K. \quad (4)$$

Recall that



is a solution for  $\mathcal{T}$ , where  $(n, m) \in \mathbb{N} \times \mathbb{N}$ :

$$\mu(n, m) = t \quad \text{if} \quad \{t\} = \rho(x_{\pi(n, m)}).$$

Note that  $\mu$  is well-defined since the formula  $\varphi_{\text{tile}}$  guarantees that  $\rho(x_{\pi(n, m)})$  is indeed a singleton. Since each first column world on  $\sigma$  is  $x_{m, m}$  for some  $m \in \mathbb{N}$ , infinitely often first column worlds satisfy  $t_0$ , and  $\pi^{-1}(m, m) = (0, m)$ , it follows that  $\mu(0, m) = t_0$  for infinitely many  $m \in \mathbb{N}$ .

Fix some  $(n, m) \in \mathbb{N} \times \mathbb{N}$  such that  $n + m$  is even. The case when  $n + m$  is odd can be handled analogously.

Let  $\mu(n, m) = t$  and  $\mu(n + 1, m) = t'$  for some  $t, t' \in T$ . We prove that  $(t, t') \in H$ . By definition, we have  $\rho(x_{\pi(n, m)}) = \{t\}$  and  $\rho(x_{\pi(n+1, m)}) = \{t'\}$ .

Note that  $\pi(n, m) = (n + m, m)$  and  $\pi(n + 1, m) = (n + m + 1, m)$  and since  $n + m$  is even, it follows by (5) that

$$(x_{\pi(n, m)}, x_{\pi(n+1, m)}) \in \llbracket L_0^{\leftrightarrow} \rrbracket_K. \quad (9)$$

Recall that  $\varphi_{\leftrightarrow}^{\uparrow}$  is defined as follows:

$$\begin{aligned} \varphi_{\leftrightarrow}^{\uparrow} = & [\Sigma^* c(a_1 b_2)^+] \bigwedge_{t \in T} t \rightarrow \left( [L_0^{\leftrightarrow}] \bigvee_{t' \in T: (t, t') \in H} t' \wedge [L_0^{\uparrow}] \bigvee_{t' \in T: (t, t') \in V} t' \right) \wedge \\ & [\Sigma^* d(a_2 b_1)^+] \bigwedge_{t \in T} t \rightarrow \left( [L_1^{\leftrightarrow}] \bigvee_{t' \in T: (t, t') \in H} t' \wedge [L_1^{\uparrow}] \bigvee_{t' \in T: (t, t') \in V} t' \right) \end{aligned}$$

By  $(x, x_{\pi(n, m)}) \in \llbracket \Sigma^* c(a_1 b_2)^+ \rrbracket_K$ , by (9), and by the definition of the formula  $\varphi_{\leftrightarrow}^{\uparrow}$ , it follows directly that  $(t, t') \in H$ .

Analogously, by applying (7), for all  $(n, m) \in \mathbb{N} \times \mathbb{N}$  such that  $\mu(n, m) = t$  and  $\mu(n, m + 1) = t'$ , we conclude that  $(t, t') \in V$ .  $\square$

Finally, we obtain the following theorem:

**Theorem 9** *Satisfiability of  $\text{PDL}(k, n)$  is  $\Sigma_1^1$ -complete.*

## References

- [1] M. J. Fischer, R. E. Ladner, Propositional dynamic logic of regular programs, J. Comput. Syst. Sci.

- [2] D. Harel, D. Kozen, J. Tiuryn, Dynamic Logic, Foundations of computing, The MIT Press, 2000.
- [3] D. Harel, A. Pnueli, J. Stavi, Propositional dynamic logic of nonregular programs, J. Comput. System Sci. 26 (2) (1983) 222–243.
- [4] C. Löding, C. Lutz, O. Serre, Propositional dynamic logic with recursive programs, J. Log. Algebr. Program. To appear.
- [5] T. Koren, A. Pnueli, There exist decidable context-free propositional dynamic logics, in: Proceedings of the Carnegie Mellon Workshop on Logic of Programs, Springer-Verlag, London, UK, 1984, pp. 290–312.
- [6] D. Harel, D. Raz, Deciding properties of nonregular programs, SIAM J. Comput. 22 (4) (1993) 857–874.
- [7] R. Alur, P. Madhusudan, Visibly pushdown languages, in: Proceedings of the 36th Annual ACM Symposium on Theory of Computing, ACM, New York, 2004, pp. 202–211 (electronic).
- [8] S. La Torre, P. Madhusudan, G. Parlato, A robust class of context-sensitive languages, in: Proceedings of the 22nd IEEE Symposium on Logic in Computer Science, IEEE, 2007.
- [9] D. Harel, A Simple Highly Undecidable Domino Problem, in: Proceedings of the Conference of Logic and Computation, 1984.